Future Trends in Software Architectures for Automotive Systems

Dr. S. Voget

Robert Bosch GmbH, Corporate Research and Development, Software Technology
P.O. Box 94 03 50, D-60461 Germany
E-mail: stefan.voget@de.bosch.com

Keywords: electronic control units, software architecture, vehicle-environment-supervision domain

Abstract

This paper focus on the future integration of software modules into vehicle electronics architecture. Based on an analysis of the challenges coming soon we present common approaches and solutions for the software architectures which are already perceptible. To validate the architecture concepts a demonstration of an example completes the contribution.

1 Introduction

The grown structures of today's electronic control units (ECUs) in the vehicles are characterized by two trends:

- An increasing number of relatively loosely coupled, mono-functional ECUs.
- An increase of the total share of software in each of the ECUs [4][5][7][8][11].

As a result of this trend, the communication between ECUs increases, too. But the networks of ECUs are oriented on a traditional grown functional division of automotive functions. Following this division the ECUs are grouped into several sub-domains, for example, power train, body, telematic, chassis, etc. The networks of the ECUs are neither standardized according to their interfaces across these sub-domain borders nor developed with respect to the interrelationships between nodes of the network.

Comparable statements are valid for the development processes. The software development processes for different ECUs have evolved based on the individual history of the sub-domains, and are currently quite divergent. In the automotive industry most of the widespread system development processes map the functional requirements one to one into software and hardware components. Software architecture models that provide an appropriate level of abstraction of the systems are often missing.

This paper is organized as follows: first, we motivate the needs for new system structures in the electronic networks of vehicles. Next, we present several surveys for activities which consider and develop future system structures. How the future software architecture division will look like is summarized in chapter 4. A demonstration of an example from the vehicle-environment-supervision domain completes the contribution. This example will be used for the validation of the architecture concepts.

2 Needs for new system structures

2.1 A Short History

A brief consideration of the history is necessary to understand the grown structures and the challenges of software development in the automotive industry. In this section we concentrate on the history of technologies in this area.

The technologies changed several times during the last 150 years. Up to World War 1 the production was determined by small series that had been mainly craft oriented. About up to the sixties, big series productions get through. In this period mechanical and first electro-mechanical products determined the technology. Up to today there had been a continuous increase of electronic in combination with more and more precision mechanics.

There will be a fourth phase of technology change in near future: the software will get of dominant importance for the costs of the vehicles. For the next ten years an increase of 10%-15% of software in the share of costs of a vehicle is forecasted every year [8]. There are two main drivers for this which are not only reasoned by the technology:

- Software makes electronic systems cheaper.
- New or improved functionality will only be possible with the help of software.

2.2 Problems With These Structures

The outlined historic survey influences the grown structure of electronic systems in the vehicles. It is one of the main reasons for the problems in the software development. These lead to delayed start of productions and an increasing number of call backs because of errors in electronic systems [14].

<u>On software level</u>

The increasing total share of software leads to high complexity and costs. This becomes more critical with non standardized development processes and without adequate networks. In addition, the incorporation of third party software increases the complexity of collaboration between companies.

An appropriate level of abstraction in the software architecture modeling and appropriate integration concepts are still missing. Current architectures do not reflect the effects of quality requirements. As a consequence, these often remain vague and unexplored. The architectures grown up by the single solution development strategy do not represent long-term solutions.

<u>On functional level</u>

Today a lot of functionality is distributed on several ECUs. E.g., indicator functionality is distributed over up to eight ECUs in high end vehicles. Furthermore, some of the future functionality is not realizable with the current loose side by side of ECUs. E.g., drive-by-wire will need a very close and safe interlocking of ECUs across different domains [6]. The traditional functional division of automotive functions is more and more crossed by the new functionalities.

In the next section we list some of these upcoming functionalities with main influence on the problems mentioned here.

2.3 Challenges because of new functionalities

As described above the present situation leads to a lot of challenges for the further development of electronic systems in vehicles. In addition several new functionalities will increase these challenges. This section contains a list of functionalities which will have main influence for the network of ECUs and the associated software architectures. The list is grouped into

- Improvement of Propulsion Technology
- Substitution of Mechanical and Hydraulic Linkage
- Intelligent Vehicles

These groups form the drivers for the evolution of the technologies presented here [1][3][12].

<u>Improvement of Propulsion Technology</u>

Technology	**Description (D) / Improvements (I)**

Improvement of Diesel Engines	**D:** Diesel engines offer a consumption advantage compared to gasoline engines due to their better thermal efficiency of about 20%. This advantage can be further improved by 15% to 20% by optimizations at the injection system, the combustion control, and the engine control system.
	I: The particle emission is reduced by high injection pressures and prevented by diesel particulate filter systems.
Hybrid Power-train	**D:** The rising electrical power demand leads to the application of the integrated starter-generator and the 42-Volt electrical system. The next step in this evolution is to use the integrated starter-generator as an electric motor. Thus a simplified hybrid drive is available. During acceleration the electric motor can support the combustion engine. During breaking it can be operated as generator to regenerate energy. In case of standstill the combustion engine can completely be switched off. Low speeds will be mastered alone with the electric motor.
	I: This leads to substantial fuel saving, in particular before red traffic lights and in stop and go traffic.

Substitution of Mechanical and Hydraulic Linkage

The substitution of mechanical and hydraulic linkages by purely electrical/electronic systems is due to the limited nature of mechanical systems in the areas of safety and comfort as well as the rapidly improving price/performance ratio of microelectronic components.

Technology	Description (D) / Improvements (I)
Electro-mechanical Valve Lift Control	**D:** The electromagnetic valve lift control replaces the camshaft. The valves are actuated by electromagnets which are controlled electronically.
	I: Gas change can be optimized such that fuel consumption is reduced or engine power is increased.
Automated Manual Transmission	**D:** An automated manual transmission operates the switching mechanism and the clutch in electric motors. A mechanical linkage between the switching level and the clutch pedal can be dropped.
	I: This influences the weight and crash behavior positively.
By-Wire Car	**D:** Components for braking, steering, shifting and other functions are controlled by electrical/electronic systems.
	I: Electronic systems can allow a reduction in overall costs during the complete life cycle of a car. The reduced space and weight necessary for electronic systems can lead to reduced fuel consumption. Electronic systems provide new concepts for remote diagnosis and maintenance, and they enable new technologies such as fuel cells and intelligent power-train control systems.
Integrated Chassis Control	**D:** Drive-by-wire chassis consist of individual corner modules that are segmented further into smart sub-modules for steering, braking, suspension and damping. These will be cross-linked via a control module with an integrated chassis controller.
	I: The objectives are to improve vehicle stability, to cut braking distance, to optimize the comfort/driving stability trade off and to harmonize steering, braking, suspension and damping systems.

Intelligent Vehicles

Technology	Description (D) / Improvements (I)
Driver Assistance Systems	**D:** Driver assistance systems support the driver in specific situations. Upcoming ultrasound and radar sensors enable new driver assistance systems like advanced Adaptive Cruise Control, Blind Spot Detection, Stop and Go Control and Parking Assistance. The introduction of video-based sensors permits the recognition of traffic signs, lanes and obstacles.

	I: Safety and comfort is improved.
Pedestrian Protection	**D:** Sensors embedded in the vehicle's front bumper detect a collision with a pedestrian's legs and raise the rear of the bonnet around 10 cm, using either gas-filled bellows or electro-mechanical activators.
	I: Pedestrian-protection systems are designed to minimize head impact in the event of a collision.
Usability	**D:** Since the entering of microelectronics the number of functions in vehicles has increased. A simple evolution of the well-known control surfaces would lead to a tide of switches and small lamps.
	I: The user interface has to present all information clearly and enable an easy and intuitively operation of the vehicle functions by different driver types, without diverting the driver.

Each and every of the listed functionalities will increase the complexity in the network of distributed hardware. The need for the control of the complexity – also in future the vehicles have to be realized – leads directly to some needs for new structures. These will be considered in the next section.

2.4 Need For New Structures

Future developments have to control and manage the increasing complexity on a network of distributed hardware. A cost minimum is not achievable on the basis of single ECU considerations. But, a cost minimum for the whole system of ECUs is necessary with the important constraint to be flexible enough for the realization of future requirements, especially with respect to the further displacement of mechanic/electric systems by software intensive systems.

To minimize costs, to win efficiency and quality it is necessary to develop new, integrated system architectures, e.g. platform based [10]. Such architectures have to be standardized to improve the collaboration between vehicle manufacturers, system suppliers and 3rd party software suppliers.

Furthermore, appropriate system architectures can only be realized by associated software architectures which reflect the requirements of the whole network of all ECUs. To enable the communication of single parts in this network a comprehensive, global software architecture framework is needed which contains a set of decisions about the used interfaces.

Several projects already work on the topic of a software architecture framework. Before we go into detail of the structures we introduce some projects dealing with these needs in the next chapter.

3 Common efforts to develop the future architectures

Several activities, initiatives, and projects work on the challenges and problems described above.

The projects where OEMs and suppliers work together will have a lot of impact on the development of system/software architecture frameworks for the future.

<u>AIL:</u> With the support of the French Under-Ministry for Industry several French OEMs and suppliers worked together in the project AEE (Architecture Electronique Embarquée). The project closed in 2001 but the results influence several subsequent activities. The mission of the project was:

- "In order to meet end customer needs in terms of quality and innovation, at the right price, and also to control the resultant growth in complexity, this project designs and validates a fast and reliable process for defining the Systems Architecture and developing the associated embedded software. This process is based on independence between hardware and software, on the use of standard methods, tools and components, and on the commitment by contracts for exchanges between the main players. The project focuses on transport applications, particularly for automobiles."[aee.inria.fr]

One of the main outputs of this project was an AIL (Architecture Implementation Language).

TITUS: As a counterpart to the French initiative some German companies developed TITUS. Originally developed for the body domain the methodology is in the meantime extended to other domains.

- "Vehicle body electronic software has reached a level of complexity and cost that methods focusing on re-use and seamless vehicle integration of networked ECUs are being heavily embraced. The TITUS methodology shows that vehicle functions can be re-used over several physical ECU networks, i.e. several vehicle models." [2]

HIS: Since 2000 some German OEMs work together in the HIS (Herstellerinitiative Software) initiative. The goal of this initiative is to develop software modules in such parts of the system which are not relevant for the competition between the OEMs:

- "The vehicle manufacturers Audi, DaimlerChrysler, BMW, Porsche, and Volkswagen introduce their joint efforts to standardize so called standard software modules for in vehicle networks. This will also include methods for determining process maturity levels, software test and software tools. The objective is to achieve common standards." [9]

ITEA-EAST/EEA: All the beforehand mentioned activities influence this project. We present it here in more detail. ITEA-EAST/EEA is an European funding project. More than 20 OEMs, suppliers, tool-suppliers, and research institutes from 4 European countries work together to…

- "… enable a proper electronic integration through definition of an open architecture. This will allow to reach hardware and software interoperability and re-use for mostly distributed hardware [Figure 1]. By developing a European wide accepted embedded electronic vehicle architecture, ITEA-EAST/EEA is a prerequisite for the implementation of these advanced features and functions." [http://www.east-eea.net]

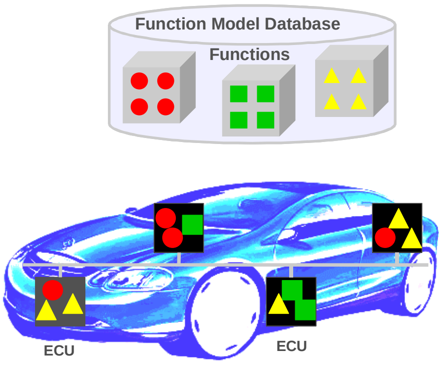

Figure 1: Various functions have to be distributed over several ECUs

The project started in 2001 and will end in 2004. The activities are based on two main columns:

- the definition and specification of a middleware and communication concept for an automotive embedded electronic architecture [Figure 2a], and
- the development and validation of architecture and functional modules where the complete development process is taken into account [Figure 2b]

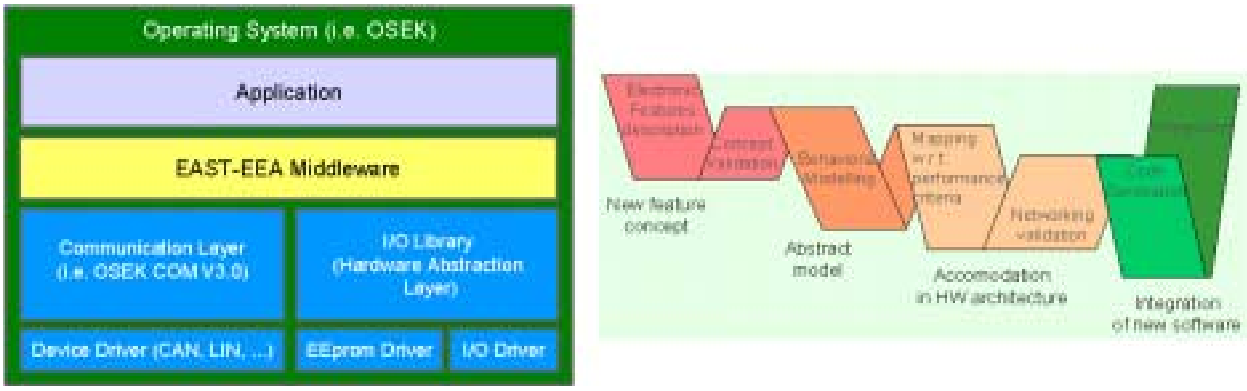

Figure 2: a) layered architecture, b) general development process

4 Trends for software architectures in the automotive domain

All those projects and initiatives have different goals in mind and follow different approaches to solve the mentioned challenges. Although several projects and companies work on the development of future software architectures with different goals a common trend is perceptible. This trend is driven by the parts of the system in which the OEMs have to differ to contrast their brand from others.

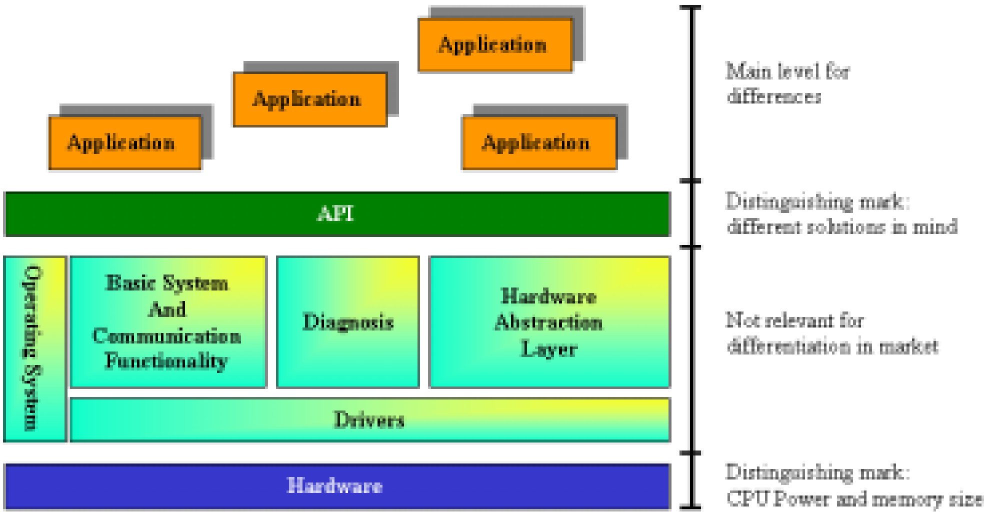

Figure 3: general layer structure

Figure 3 shows the general structure of an ECU which is forecasted as a standard in near future ECUs. On the hardware level the OEMs will differ from each other. The main values for differentiation are the number of ECUs, the CPU-power, and quantity of memory in the hardware. A vehicle in the luxury class will have more ECUs and ECUs with more CPU-power than a car in a compact class.

The next layer – sometimes called EBS (embedded basic software) – consists of the operating system, drivers, basic system and communication functionality, diagnosis services, and hardware abstraction layer. These parts of the software are not relevant for the individual differentiation of an OEM in market. Therefore, OEMs as well as suppliers are interested to define and to use standards in this area. The details given in Figure 3 for the EBS part are more or less common for all the mentioned initiatives and projects.

A slightly different situation can be found on the next higher layer, the API. This layer is also not relevant for the differentiation in market. But for this part of the system all initiatives and projects have different solutions in mind. Some projects put only a thin layer with interfaces in this part. Others add several set of services (e.g. notification services, mode management, ...) which can be used by all applications positioned above this layer. The thickest versions are constructed by extensive runtime environments which will be automatically generated and embed the applications and their relationships.

The application level is the part of the software which is the most relevant part for the differentiation in market. Today there are no efforts for common solutions or standardizations in this layer noticeable. From a supplier point of view this situation prevents the suppliers from cost efficient development by reusable software modules [13]. In future the establishment of - at least - common component models will be decisive for the success of cost efficient application software modules from the suppliers.

5 Example: portable sensor software

Currently there are several pilot projects which experiment on the concepts presented in the last chapter. In the ITEA-EAST/EEA project Bosch builds up a vehicle to validate the architecture concepts. The vehicle-environment-supervision domain is used to show the possibilities which will be enabled by distributable application software modules in future.

The vehicle-environment-supervision domain can be characterized as a family of applications that are based on sensors installed around the vehicle to monitor its local environment. The sensors enable either the realization or enhancement of several applications, like park assistant, control of reversible hold back systems (e.g. seatbelt tensioner), conditioning of airbags, and blind spot detection. To realize these applications several sensors of different technologies – e.g. ultrasound sensors or microwave sensors – are used [Figure 4].

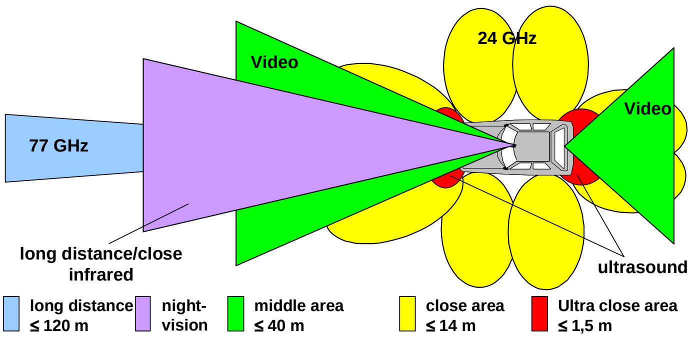

Figure 4: Overview about distance areas and associated sensor technologies

A processing in several steps leads to a description of the environment and the obstacles in these areas. The necessary software has to be distributed on several sensors and ECUs. The hardware equipments differ from one OEM to another. But some constraints on the used hardware and topology of ECUs are reasoned in the requirements of the applications on the hardware, e.g. a pre-crash is not realizable with ultrasound sensors.

In the ITEA-EAST/EEA project parts of a park-assistant will be realized independent from the underlying sensor technology – subject to all constraints – and the ECU hardware. As a second scenario the upgrade of sensor processing software via CAN Bus from a telematic unit will be realized [Figure 5]. Both scenarios together will show that a flexible distribution of an application is possible in (at least) the body domain. In addition, the limits of such a procedure are tested out. The realization enables the inclusion of the park-assistant software modules in function-libraries. Several OEMs plan to install such data bases. This will show how the integration into company specific libraries could look like.

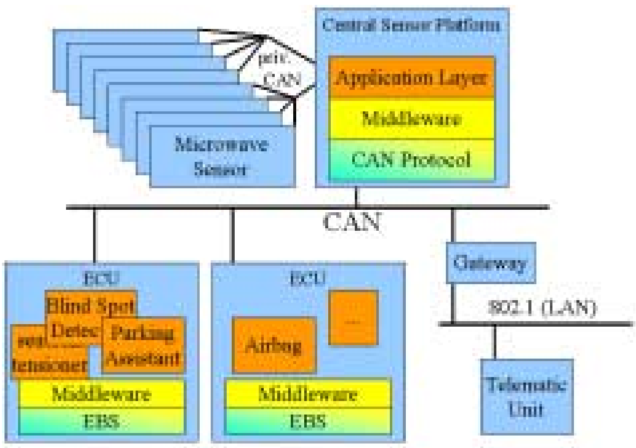

Figure 5: network structure for the validation

6 Conclusions

We have shown that the electronic networks in the automotive domain will change in near future. This will take place on system- as well as on software level. For specific parts of the software architectures common approaches and solutions are already perceptible. They have a good chance to evolve into standards.

But the enormous grows of new functionalities to be included shows the complexity of the necessary solutions. Flexibility with portable software modules is not reachable with cost optimal solutions. Today the two quality attributes “portability” and “cost optimal” are often considered separately. In addition the quality attribute “portability” is often contrary to safety requirements. Future considerations have to take into account a sufficient compromise between all relevant quality attributes.

Acknowledgements

This work was partially funded by the ITEA project No. 00009 EAST/EEA which is part of the EUREKA program. We would like to thank A. Lehn for his assistance in the section ”Challenges because of new functionalities”.

References

[1] R. Beck et al.; “Backbone Architecture: Evolution of the Central Gateway to an Integrated Communication System“ Conference Proceedings - Elektronik im Kraftfahrzeug, Baden-Baden 2001, VDI Berichte 1646.

[2] U. Freund, A. Burst; “Titus – A Graphical Design Methodology for Embedded Automotive Software” Workshop on Object-oriented Modelling of Embedded Real-time Systems, OMER2, 2000.

[3] P.E.H. Hoffman, Thomas Thurner; „Neue Elektrik/Elektronik Architekturansätze” Conference Proceedings - Elektronik im Kraftfahrzeug, Baden-Baden 2001, VDI Berichte 1646.

[4] ITEA; “Technology Roadmap on Software Intensive Systems”, ITEA Office Association, Eindhoven, March 2001.

[5] K. Jost; “Electronics demand to grow nearly 7% annually” in Automotive Engineering International, September 2001.

[6] K. Jost; “From fly-by-wire to drive-by-wire” in Automotive Engineering International, September 2001.

[7] ATKerney; “Software-betriebene Fahrzeugsysteme bestimmen die Zukunft des Automobils” Study, 2001.

[8] McKinsey&Company; “Automotive software: A battle for value” Study, 2002.

[9] K. Lange et. al.; „Herstellerinitiative Software“ VDI Conference for electronic in vehicles, Baden-Baden, 2001.

[10] H. Postema; “Platform Based Product Development” In F. van der Linden (Ed.): Software Product-Family Engineering; Lecture Notes on Computer Science (LNCS) Vol. 2290; Springer; Berlin 2002.

[11] D. Sallee, R. Bannatyne; “Trends in Advanced Chassis Control” in Automotive Engineering International, September 2001.

[12] P. Steiner, F. Schmidt; „Requirements and System Architecture of future Body Electronics“ Conference Proceedings “Elektronik im Kraftfahrzeug”, Baden-Baden 2001, VDI Berichte 1646.

[13] S. Voget; “Product Line Variability in the Automotive Domain” In Conference Proceedings: International Colloquium of the Sonderforschungsbereich 501; University of Kaiserslautern; 2003.

[14] S. Voget, M. Becker; "Establishing a Software Product Line in an Immature Domain" In G. Chastek (Ed.) - Software Product Lines, Springer2002.